Selection

Decision Making and IF Statements

Selection

- Allows your program to make decisions
  - Gives the user choices
  - Follows different paths depending on what choices are made
- Through the IF structure, your programs can branch to a section of code or just skip it.

Operator	Meaning	Sample Condition	Evaluates to…..
= =	Equal to	5 == 5	True
! =	Not equal to	5 != 3	True
>	Greater than	5 > 8	False
<	Less than	4 < 3	False
>=	Greater than or equal to	3 >= 3	True
<=	Less than or equal to	5 <=10	True

EXAMPLE

- To pass this course you must obtain at least a 50% or higher. Print a message to the screen that notifies the user if they pass or fail.

Python Code

Will this code work? What will be the expected I/O?

```
mark = input("Enter your mark\n")
if mark >= 50:
    print ("You pass!!")
else:
    print("You fail!")
```

Actual Output is …

```
if mark >= 50:
TypeError: '>=' not supported between instances of 'str' and 'int'
```

Corrected Python Code

```
mark = int(input("Enter your mark\n"))
if mark >= 50:
    print ("You pass!!")
else:
    print("You fail!")
```

BLOCKS

- For the following segment of code:

```
mark = int (input("Enter your mark\n"))
if mark >= 50:
    print ("You pass!!")
else:
    print("You fail!")
```

- Notice that the second and fourth lines are indented

BLOCKS Cont'd

- By indenting the line, it becomes a block.
- A block is one or more consecutive lines indented by the same amount
- If the condition in an if statement is true the entire block is executed
  - If the condition is false the block is bypassed

What if there were more than two options besides pass and fail? What if we wanted to check if you made the honour roll?

- There are three possibilities for checking different possibilities in the selection structure:

if-elif-else

if–elif-else

- When you have one variable to compare to a bunch of different values

- For example, if you obtain an 80% or above in this class, you will receive honours. A mark of 50 or greater will earn you a pass. Otherwise you fail.

Example #1: Will this work as expected? If not, what kind of error is this?

```
mark = int(input("Enter your mark\n"))
if mark >= 50:
    print ("You passed the class")
elif mark >=80:
    print ("You're on the honour roll!")
else:
    print("You failed!")
print("The program is now over")
```

Corrected Code.

The logic error has been fixed.

```
mark = int(input("Enter your mark\n"))
if mark >= 80:
    print ("You're on the honour roll!")
elif mark >=50:
    print ("You passed the class")
else:
    print("You failed!")
print("The program is now over")
```

String Comparisons

- To compare string values, we compare the strings based on ASCII
- ASCII (American Standard Code for Information Interchange)
  - ASCII is a code for representing English characters as numbers.
  - Therefore each letter is represented by a number and we are comparing those numbers

The first 32 characters in ASCII are set aside for communications and printer control.

ASCII chart

American Standard Code for Information Interchange

33	!		49	1		65	A		81	Q		97	a		113	q	
34	“		50	2		66	B		82	R		98	b		114	r	
35	#		51	3		67	C		83	S		99	c		115	s	
36	$		52	4		68	D		84	T		100	d		116	t	
37	%		53	5		69	E		85	U		101	e		117	u	
38	&		54	6		70	F		86	V		102	f		118	v	
39	‘		55	7		71	G		87	W		103	g		119	w	
40	(		56	8		72	H		88	X		104	h		120	x	
41	)		57	9		73	I		89	Y		105	i		121	y	
42	*		58	:		74	J		90	Z		106	j		122	z	
43	+		59	;		75	K		91	[		107	k		123	{	
44	,		60	<		76	L		92	\		108	l		124	|	
45	-		61	=		77	M		93	]		109	m		125	}	
46	.		62	>		78	N		94	^		110	n		126	~	
47	/		63	?		79	O		95	_		111	o		127	DEL	
48	0		64	@		80	P		96	`		112	p				

String Comparison Example # 1

- When comparing strings, Python will compare the two strings from left to right character to character.

- e.g. “brAthwaite” == “brathwaite”
  - Python will compare the first character of each string
  - Python will determine if ‘b’==‘b’, ‘r’ == ‘r’ and so forth
  - However ‘A’ < ‘a’. Therefore the condition is false
  - ‘A’ in ASCII is 65 and ‘a’ is 97

String Comparison Example # 2

- We can also determine if one string is greater than another.
- Consider the following condition:

  “hAROLD” > “Harold”
- Since Python compares the string one character at a time, Python will conclude that ‘h’ > ‘H’.
  - Therefore the condition ‘hAROLD’ > ‘Harold’ is true regardless of the characters that follow.

Today's Class

- Work on new Decision and Selection Exercises on the GC
- Complete any outstanding Python exercises